User's Manual

32

R32C/102 Group

User's Manual: DSP

RENESAS MCU

M16C Family / R32C/100 Series

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest informaton published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev. 1.00 May 2011

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

About This Manual

1. Purpose and Target User

This manual is designed to be read primarily by application developers who have an understanding of this microcomputer (MCU) including its hardware functions and electrical characteristics. The user should have a basic understanding of electric circuits, logic circuits and, MCUs.

Carefully read all notes in this document prior to use. Notes are found throughout each chapter.

The Revision History at the end of this manual summarizes primary modifications and additions to the previous versions. For details, please refer to the relative chapters or sections of this manual.

The R32C/102 Group includes the documents listed below. Verify this manual is the latest version by visiting the Renesas Electronics website.

Type of Document	Contents	Document Name	Document Number
Datasheet	Overview of Hardware and Electrical Characteristics	R32C/102 Group Datasheet	REJ03B0300-0101
User's Manual: Hardware	Specifications and detailed descriptions of: -pin layout -memory map -peripherals -electrical characteristics -timing characteristics Refer to the Application Manual for peripheral usage.	R32C/102 Group User's Manual: Hardware	REJ09B0578-0101
User's Manual: DSP	Specifications and detailed descriptions (memory map and peripherals) of the DSP embedded in the DAP of this MCU	R32C/102 Group User's Manual: DSP	This publication
User's Manual: Software/Software Manual	Descriptions of instruction set	R32C/100 Series Software Manual	REJ09B0267-0100
Application Note	-Usages -Applications -Sample programs -Programing technics using Assembly language or C programming language	Available on the Renesas Electronics website.	
Renesas Technical Update	Bulletins on product specifications, documents, etc.		

2. Numbers and Symbols

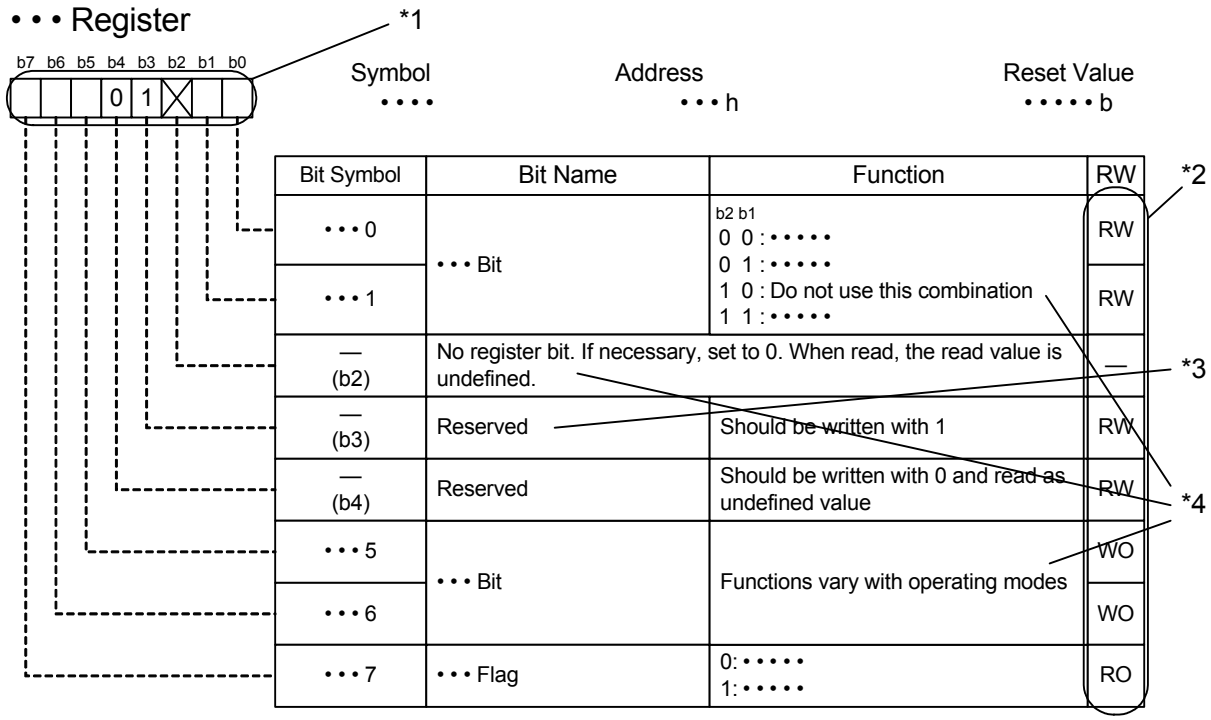
The following explains the denotations used in this manual for registers, bits, pins and various numbers.

(1) Registers, bits, and pins
Registers, bits, and pins are indicated by symbols. Each symbol has a register/bit/pin identifier after the symbol.
Example: PM03 bit in the PM0 register
P3_5 pin, VCC pin

(2) Numbers
A binary number has the suffix “b” except for a 1-bit value.
A hexadecimal number has the suffix “h”.
A decimal number has no suffix.
Example: Binary notation: 11b
Hexadecimal notation: EFA0h
Decimal notation: 1234

3. Registers

The following illustration describes registers used throughout this manual.



*1
Blank box: Set this bit to 0 or 1 according to the function.
0: Set this bit to 0.
1: Set this bit to 1.
X: Nothing is assigned to this bit.

*2
RW: Read and write
RO: Read only
WO: Write only (the read value is undefined)
—: Not applicable

*3
- Reserved bit: This bit field is reserved. Set this bit to a specified value. For RW bits, the written value is read unless otherwise noted.

*4
- No register bit(s): No register bit(s) is/are assigned to this field. If necessary, set to 0 for possible future implementation.
- Do not use this combination: Proper operation is not guaranteed when this value is set.
- Functions vary with operating modes: Functions vary with peripheral operating modes. Refer to register illustrations of the respective mode.

4. Abbreviations and Acronyms

The following acronyms and terms are used throughout this manual.

Abbreviation/Acronym	Meaning
ACIA	Asynchronous Communication Interface Adapter
bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
GSM	Global System for Mobile Communications
Hi-Z	High Impedance
IEBus	Inter Equipment Bus
I/O	Input/Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connection
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

All trademarks and registered trademarks are the property of their respective owners.

TABLE OF CONTENTS

1. Overview 1
1.1 Performance Overview ... 1
2. DAP Control Registers 3
3. DSP 4
4. Memory and DSP Peripheral Function Registers 5
4.1 Memory ... 5
4.2 DSP Peripheral Function Registers ... 6
4.3 Access Cycles ... 6
5. Serial Audio Interface 7
5.1 Overview ... 7
5.2 Registers ... 9
5.2.1 Audio Data Transmit/Receive Buffer Registers ... 9
5.3 Audio Data Transmission/Reception ... 11
5.3.1 Reception ... 11
5.3.2 Transmission ... 12
6. Synchronous Serial Interface 13
6.1 Overview ... 13
6.2 Registers ... 14
6.3 Data Transmission and Reception ... 17
6.3.1 Data Transmission ... 17
6.3.2 Data Reception ... 18
6.3.3 Overrun Error ... 19
7. Interrupt Controller 20
7.1 Overview ... 20
7.2 Registers ... 22
7.3 Operation Sequence ... 24
7.3.1 Interrupt Setting Procedure ... 24
7.3.2 Single Interrupt ... 24
7.3.3 Multiple Interrupts ... 25
8. General Purpose I/O Ports 26
8.1 Overview ... 26
8.2 Register ... 27
8.3 Input/Output Setting ... 28
8.3.1 Input Setting ... 28
8.3.2 Output Setting ... 29

9. JTAG Interface 30

10. Bootloader 31

10.1 Overview 31

10.2 Procedure on Starting DSP 32

10.3 DSP Program Transfer 33

10.3.1 Data Packet 33

10.3.2 Checksum 34

10.4 Transmit Data Formats 35

10.5 Communication Authorization Code and Checksum Reception 35

1. Overview

The R32C/102 Group is a system in package (SiP) incorporating the R32C/100 Series CPU core embedded MCU and the CoolFlux DSP (digital signal processor) embedded digital audio processor (DAP) in one package.
The MCU and the DAP operate independently enabling the CPU to control devices and the DSP to control sound fields in parallel.
The MCU is connected to the DAP by two channels of serial interface to control the DAP and provide communication between the CPU and DSP.
DSP programs are also transferred from the MCU ROM to the DAP RAM using the serial interface.
This manual describes specifications of the DAP peripheral functions controlled by the DSP, and program transfer from the MCU to DAP.

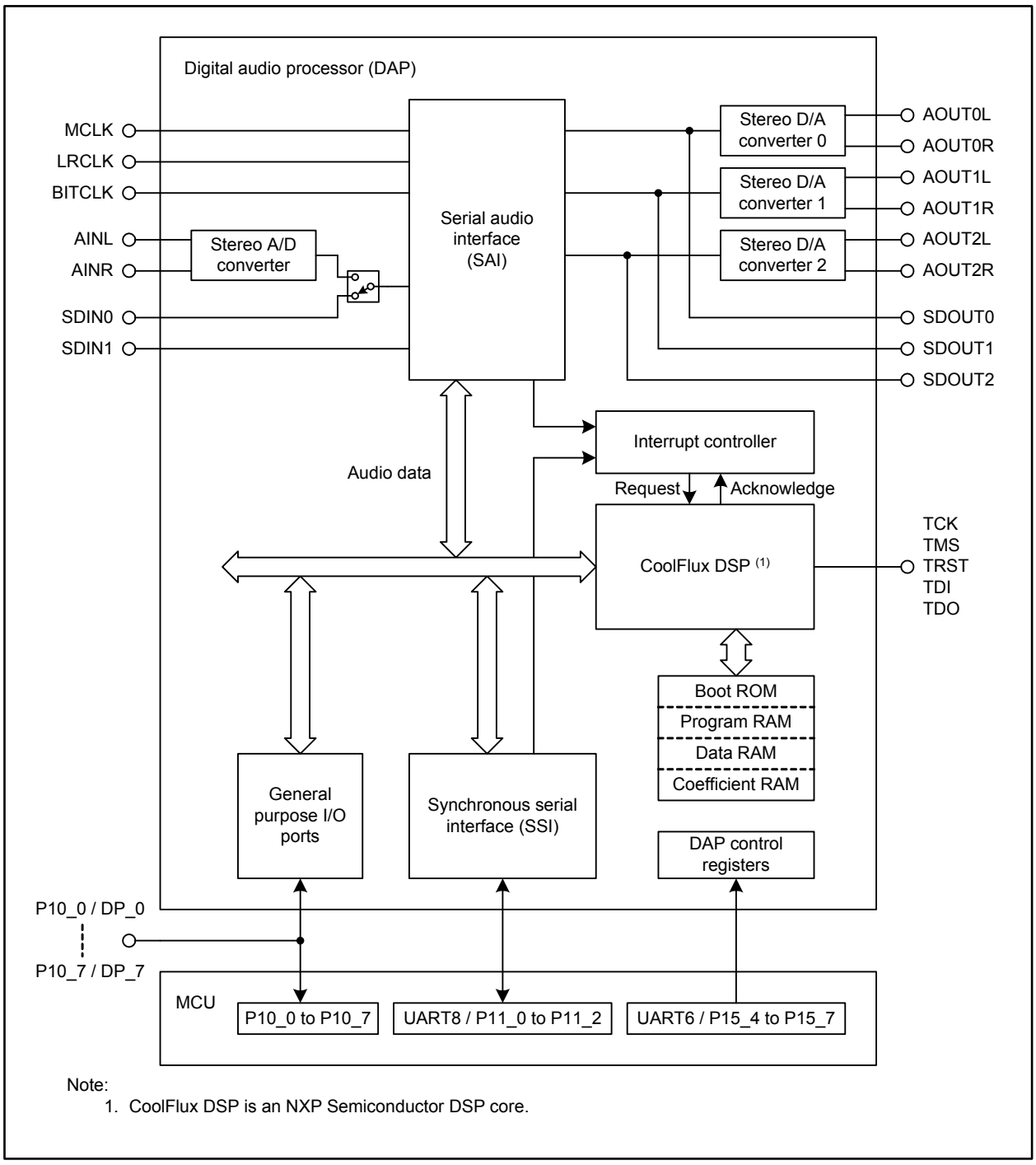
1.1 Performance Overview

Table 1.1 lists the specifications, and Figure 1.1 shows a block diagram.

Table 1.1 Specifications

Item		Specification	
		R5J64026LPFE	R5J64026PFE/R5J64029PFE
DSP		CoolFlux DSP core	
	Performance	1024 step/fs	1536 step/fs
	Operating Frequency (DSP clock)	1024 fs 45.158 MHz (fs = 44.1 kHz) 49.152 MHz (fs = 48.0 kHz)	1536 fs 67.737 MHz (fs = 44.1 kHz) 73.728 MHz (fs = 48.0 kHz)
	Arithmetic Precision	24 bits Multiply-accumulate unit: 24-bit × 24-bit + 56-bit → 56 bits (2 circuits)	
Memory		• Boot ROM • Program RAM: 1,536 words of 32 bits • Data RAM: 4,096 words of 24 bits • Coefficient RAM: 512 words of 24 bits	• Boot ROM • Program RAM: 3,072 words of 32 bits • Data RAM: 12,288 words of 24 bits • Coefficient RAM: 1,792 words of 24 bits
Audio Interface		Input: 2 channels Output: 3 channels	
Synchronous Serial Interface		1 channel (exclusively used for the communication with the CPU)	
Interrupt Controller		Interrupt sources: 2 • SAI interrupt • SSI interrupt Select either the SSI transmit buffer empty interrupt or SSI reception complete interrupt	
General Purpose I/O Port		8 ports (connected to MCU port P10 in a wired-OR configuration)	

The DSP can access the program RAM, data RAM, coefficient RAM, and DSP peripheral function registers. The bootloader is stored in the boot ROM and transfers the DSP programs and data sent from the CPU to the RAMs.
In the serial audio interface (SAI), serial audio data input to the DAP is converted to DSP readable data. The DSP processed data is converted to serial audio data.
Interrupt sources generated in the SAI and SSI are processed in the DAP's internal interrupt controller, and the interrupt request signals are input to the DSP.



Note:
1. CoolFlux DSP is an NXP Semiconductor DSP core.

Figure 1.1 DAP Block Diagram

2. DAP Control Registers

DAP control registers are accessed by the CPU and used to set the DSP operating conditions.
Table 2.1 lists only the DAP control registers required for controlling the DSP.
Refer to the “R32C/102 Group User’s Manual: Hardware” for details on the DAP control registers.

Table 2.1 DAP Control Registers Required for Controlling the DSP

Register	Symbol	Function
Audio Interface Configuration Register	AIFC	Selecting serial audio data format
DSP Control Register	DSPC	Selecting DSP operating mode
CODEC Data Selector Control Register	CDSC	Switching audio input signal
DAP I/O Pin Control Register 2	DPC2	Controlling general purpose I/O ports
DAP I/O Pin Control Register 3	DPC3	Controlling general purpose I/O ports
DAP I/O Pin Control Register 4	DPC4	Setting environment for serial communication
DAP I/O Pin Control Register 7	DPC7	Controlling general purpose I/O ports
DAP I/O Pin Control Register 8	DPC8	Controlling general purpose I/O ports
DAP I/O Pin Control Register 10	DPC10	Setting environment for serial communication
DAP Input Signal Control Register 0	DIC0	Controlling general purpose I/O ports

3. DSP

Figure 3.1 shows a block diagram of the DSP.

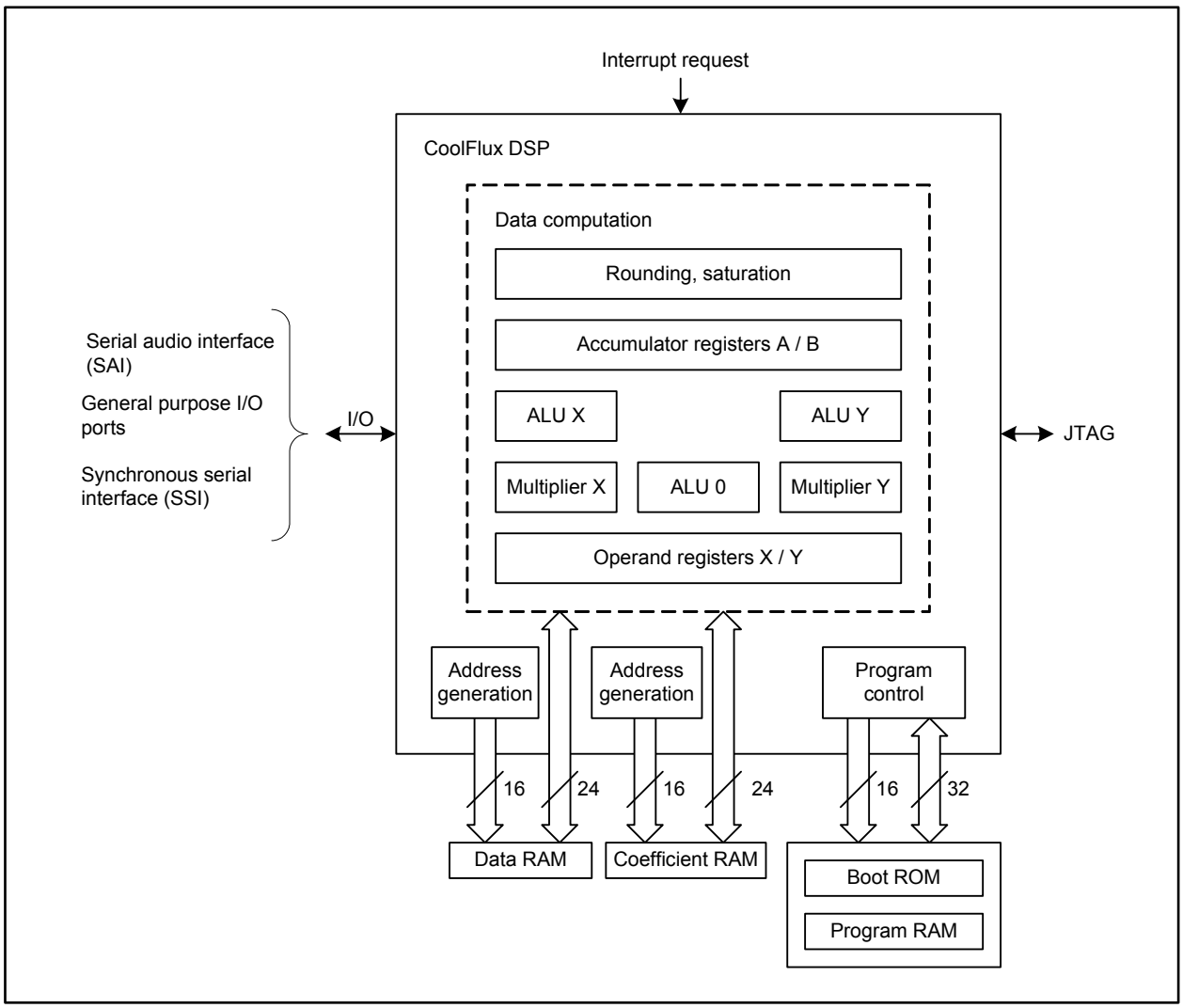


Figure 3.1 DSP Block Diagram

4. Memory and DSP Peripheral Function Registers

4.1 Memory

The DSP uses program memory, data RAM, and coefficient RAM. The program memory contains a boot ROM and program RAM. The target memory switches to the boot ROM in program transfer mode, and program RAM in program execution mode.
Table 4.1 lists memory specifications.

Table 4.1 Memory Size

Memory	Bits Per Word	Memory Size	
		R5J64026LPFE	R5J64026PFE/ R5J64029PFE
Program RAM	32 bits	1,536 words	3,072 words
Data RAM	24 bits	4,096 words	12,288 words
Coefficient RAM	24 bits	512 words	1,792 words

4.2 DSP Peripheral Function Registers

Use the DSP peripheral function registers mapped in I/O space to configure peripheral functions such as the audio interface and serial interface. Table 4.2 lists the DSP peripheral function registers. Refer to "CoolFlux DSP Assembly Programmer's Manual" or "CoolFlux DSP C Programmer's Manual" for details on accessing the I/O space.

Table 4.2 DSP Peripheral Function Registers

Address	Register	Symbol	Reset Value
0000h	Interrupt Control Register 0	ICR0	000000h
0001h	Interrupt Control Register 1	ICR1	000000h
0002h	Interrupt Control Register 2	ICR2	000000h
0007h	DSP I/O Register	DIOR	000000h
0010h	Audio Receive Buffer Register 0 L	A0LRB	XXXXXXh
0011h	Audio Receive Buffer Register 0 R	A0RRB	XXXXXXh
0012h	Audio Transmit Buffer Register 0 L	A0LTB	000000h
0013h	Audio Transmit Buffer Register 0 R	A0RTB	000000h
0014h	Audio Transmit Buffer Register 1 L	A1LTB	000000h
0015h	Audio Transmit Buffer Register 1 R	A1RTB	000000h
0016h	Audio Transmit Buffer Register 2 L	A2LTB	000000h
0017h	Audio Transmit Buffer Register 2 R	A2RTB	000000h
0018h	Audio Receive Buffer Register 1 L	A1LRB	XXXXXXh
0019h	Audio Receive Buffer Register 1 R	A1RRB	XXXXXXh
0020h	Serial Interface Setting Register	SIR	000000h
0022h	Serial Control Register	SCR	000000h
0023h	Transmit Buffer Register	TBR	0000XXh
0024h	Serial Status Register	SSR	000084h
0025h	Receive Buffer Register	RBR	0000XXh

X: Undefined

4.3 Access Cycles

Table 4.3 lists the memory types and numbers of cycles to access the DSP peripheral function registers.

Table 4.3 Access Cycles

Memory Type	Access Cycles
Program RAM	1
Data RAM	1
Coefficient RAM	1
DSP Peripheral Function Registers (1)	2

Note:
1. These registers cannot be accessed continuously using indirect addressing.

5. Serial Audio Interface

5.1 Overview

The serial audio interface (SAI) connects the DSP to the A/D and D/A converters. There are two channels of SAI for input and three channels for output, and they are all synchronized with the LRCLK and BITCLK. Each channel has a data buffer for the L channel and R channel. Serial audio data is input/output via audio buffer registers.
Table 5.1 lists the specifications, and Figure 5.1 shows the serial audio interface and its peripherals.

Table 5.1 Specifications

Item	Specification
Data Format	Selectable from the following: Left-justified format, Right-justified format, or I^2S format
Bit Order	Fixed to MSB first
Data Length	Selectable from the following: 24 bits, 20 bits, or 16 bits

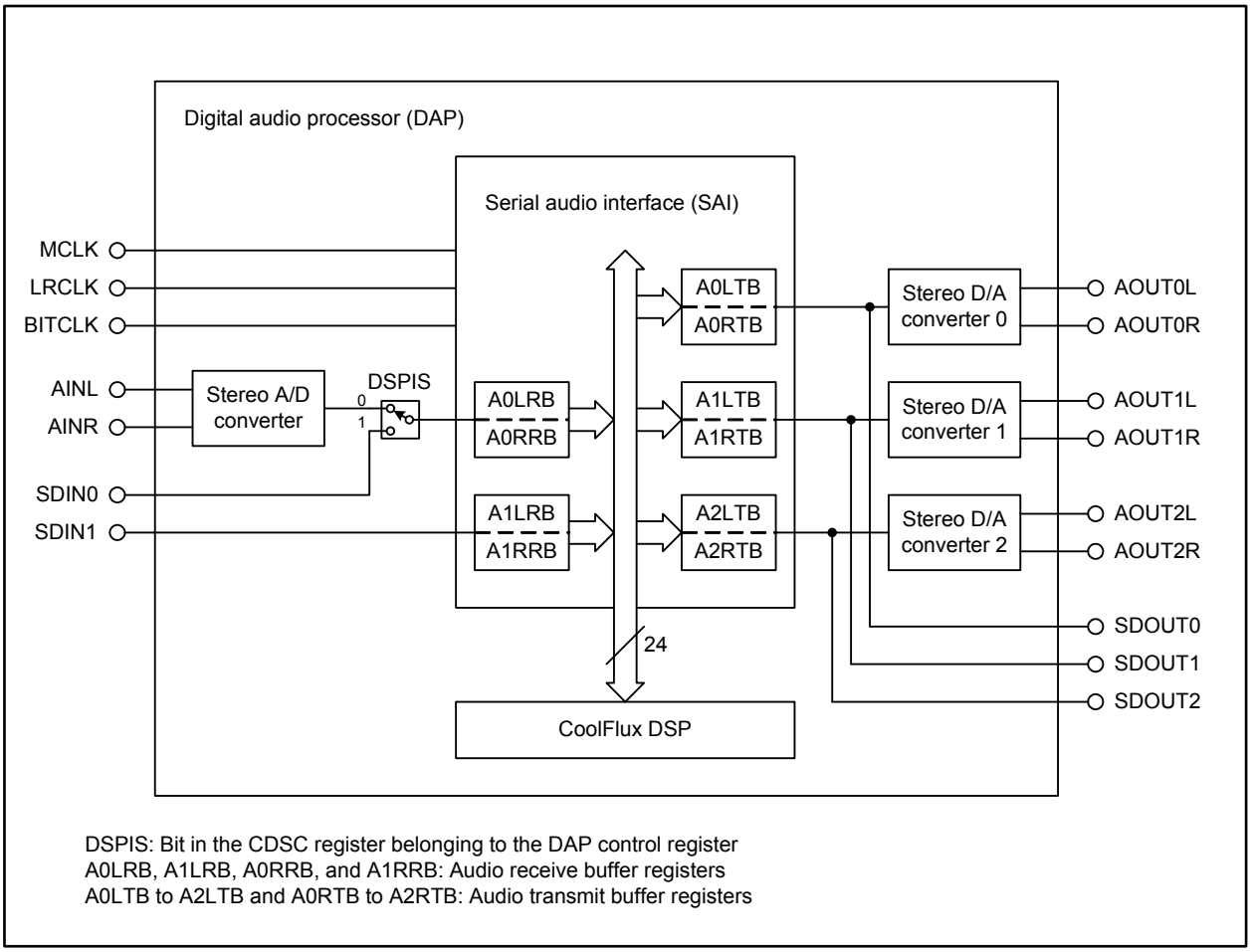


Figure 5.1 Block Diagram of the Serial Audio Interface and Peripherals

Set bits FMT3 to FMT0 in the AIFC register belonging to the DAP control register to select the data format and data length.
Figure 5.2 shows each format.

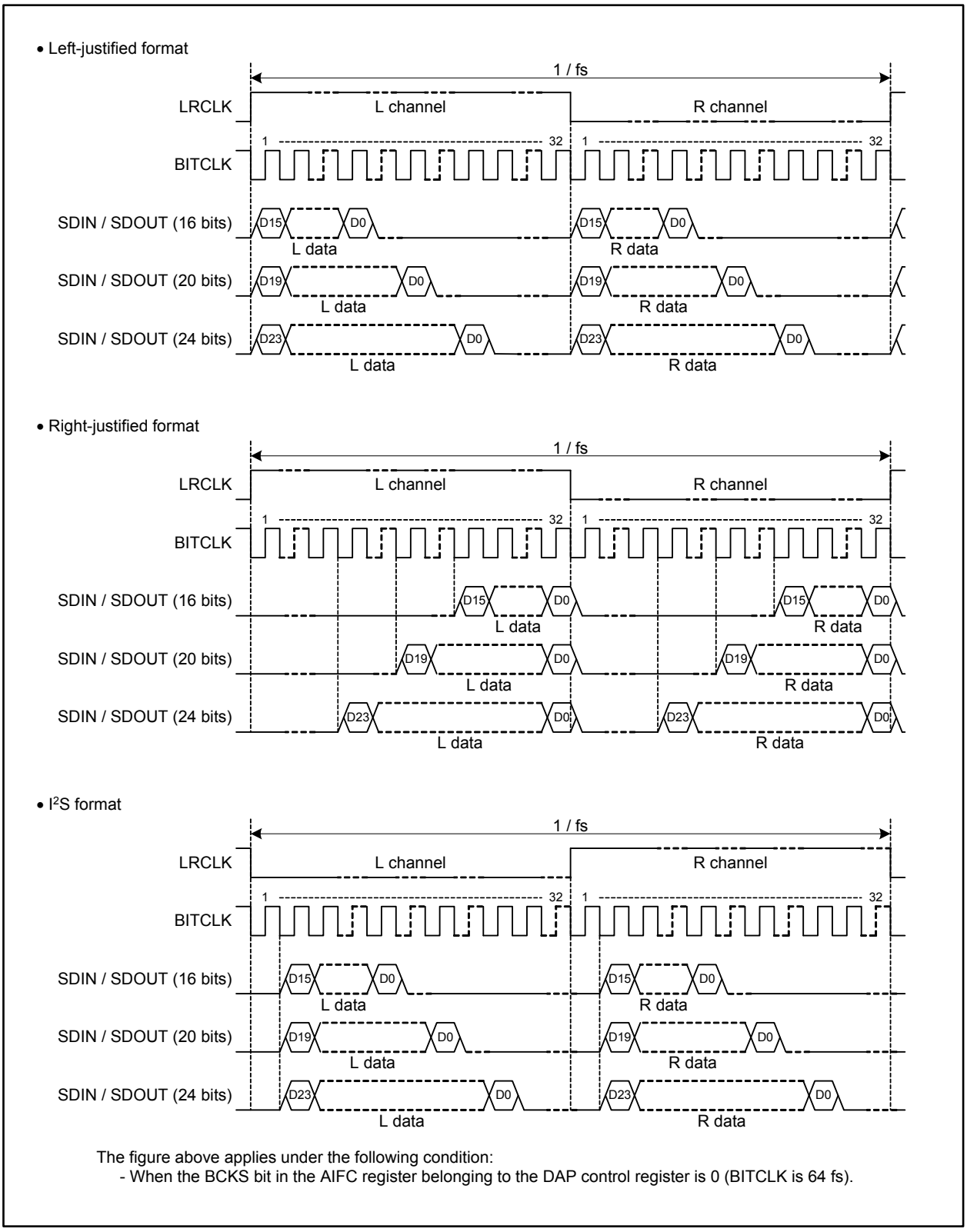


Figure 5.2 Data Formats

5.2 Registers

5.2.1 Audio Data Transmit/Receive Buffer Registers

The DSP inputs and outputs audio data via buffer registers. Set bits FMT3 to FMT0 in the AIFC register belonging to the DAP control register to select the data length. Figure 5.3 shows the bit configuration of each buffer register, and Figure 5.4 to Figure 5.7 show the buffer registers.

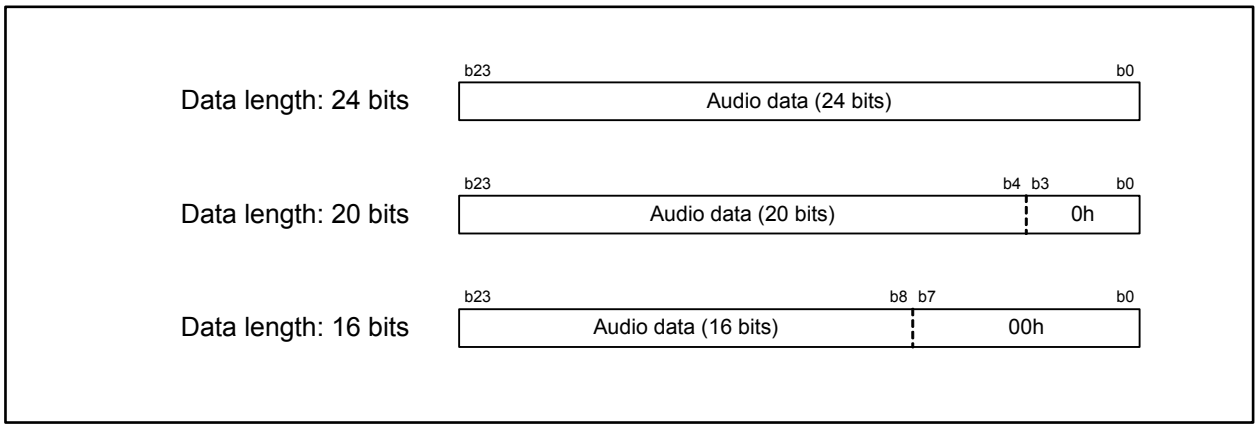
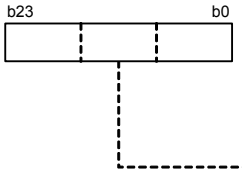


Figure 5.3 Bit Configurations of the Buffer Registers

The 4 lower bits in the 20-bit data and the 8 lower bits in the 16-bit data are all read as 0 from the receive buffer registers. They are all transmitted as 0 regardless of the values written to the transmit buffer registers.

Audio Receive Buffer Register Ln (n = 0, 1)

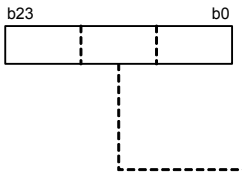


Symbol	Address	Reset Value
A0LRB, A1LRB	0010h, 0018h	Undefined

Bit Symbol	Function	RW
— (b23-b0)	L channel receive audio data	RO

Figure 5.4 Registers A0LRB and A1LRB

Audio Receive Buffer Register Rn (n = 0, 1)



Symbol	Address	Reset Value
A0RRB, A1RRB	0011h, 0019h	Undefined

Bit Symbol	Function	RW
— (b23-b0)	R channel receive audio data	RO

Figure 5.5 Registers A0RRB and A1RRB

Audio Transmit Buffer Register Ln (n = 0 to 2)

b23 b0

Symbol	Address	Reset Value
A0LTB, A1LTB, A2LTB	0012h, 0014h, 0016h	00 0000h

Bit Symbol	Function	RW
— (b23-b0)	L channel transmit audio data	WO

Figure 5.6 Registers A0LTB to A2LTB

Audio Transmit Buffer Register Rn (n = 0 to 2)

b23 b0

Symbol	Address	Reset Value
A0RTB, A1RTB, A2RTB	0013h, 0015h, 0017h	00 0000h

Bit Symbol	Function	RW
— (b23-b0)	R channel transmit audio data	WO

Figure 5.7 Registers A0RTB to A2RTB

5.3 Audio Data Transmission/Reception

5.3.1 Reception

Figure 5.8 shows an operation example when receiving serial audio data.

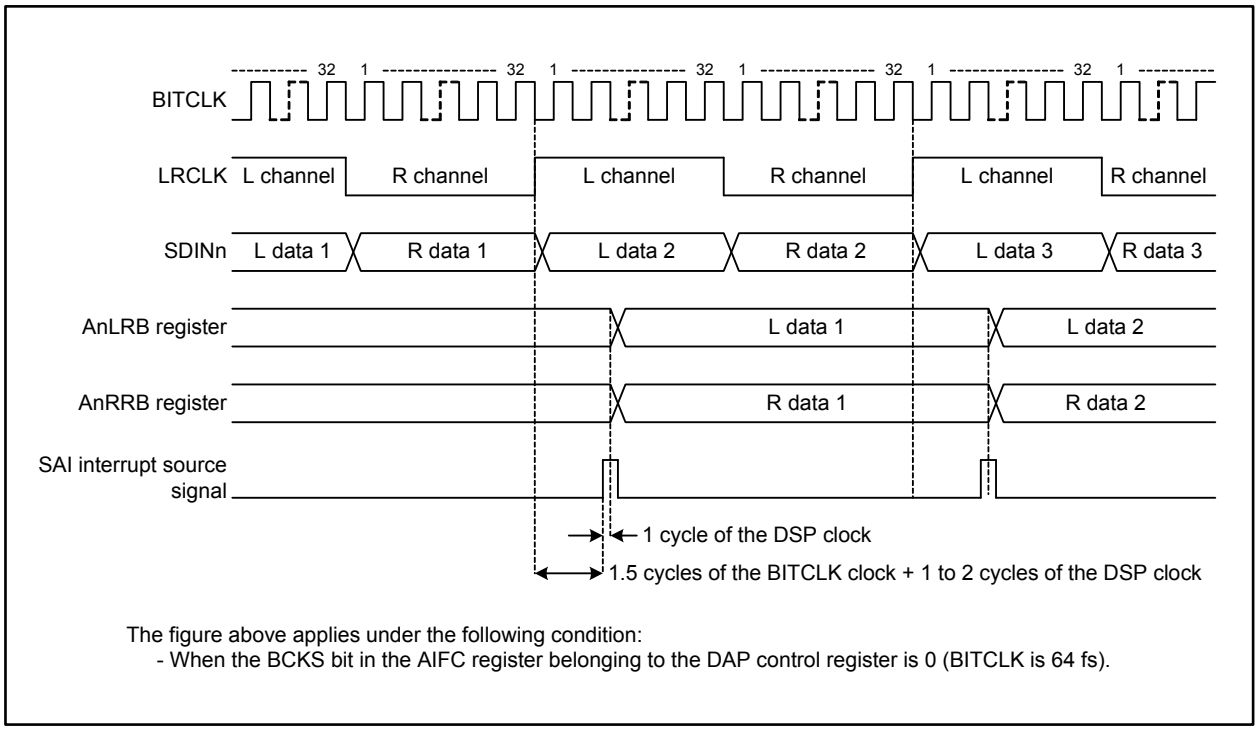

32
1
BITCLK
LRCLK
L channel
R channel
SDINn
L data 1
R data 1
L data 2
R data 2
L data 3
R data 3
AnLRB register
AnRRB register
SAI interrupt source signal
1 cycle of the DSP clock
1.5 cycles of the BITCLK clock + 1 to 2 cycles of the DSP clock
The figure above applies under the following condition:
- When the BCKS bit in the AIFC register belonging to the DAP control register is 0 (BITCLK is 64 fs).

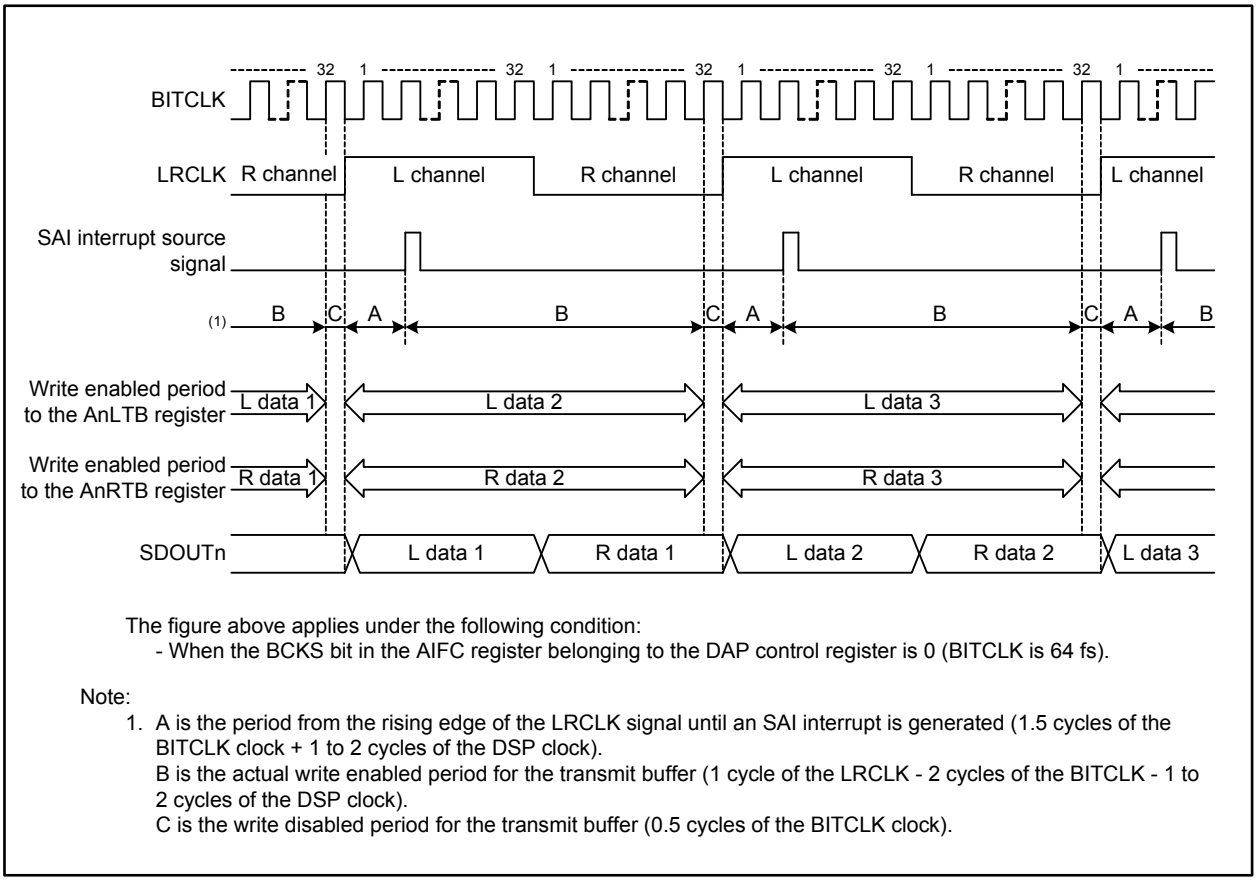
Figure 5.8 Operation Example When Receiving Audio Data in Left-justified Format (n = 0, 1)

After the L channel and R channel have successively completed receiving serial audio data, the data is transferred to registers AnLRB and AnRRB, respectively. Then an SAI interrupt source signal is generated.

The SAI interrupt source signal is input to the DAP's internal interrupt controller. Refer to 7. "Interrupt Controller" for details on interrupts.

5.3.2 Transmission

Figure 5.9 shows an operation example when transmitting audio data.



The figure above applies under the following condition:
- When the BCKS bit in the AIFC register belonging to the DAP control register is 0 (BITCLK is 64 fs).

Note:
1. A is the period from the rising edge of the LRCLK signal until an SAI interrupt is generated (1.5 cycles of the BITCLK clock + 1 to 2 cycles of the DSP clock).
   B is the actual write enabled period for the transmit buffer (1 cycle of the LRCLK - 2 cycles of the BITCLK - 1 to 2 cycles of the DSP clock).
   C is the write disabled period for the transmit buffer (0.5 cycles of the BITCLK clock).

Figure 5.9 Operation Example When Transmitting Audio Data in Left-justified Format (n = 0 to 2)

The audio data written to registers AnLTB and AnRTB is output as serial audio data at the next frame. As shown in Figure 5.9, data should not be written to registers AnLTB and AnRTB during period C since it is a preparation period for the serial audio data to be output. Although writing to registers AiLTB and AiRTB is enabled during periods A and B, write data during period B using the SAI interrupt since the DSP cannot detect the timing of the LRCLK rising edge.

6. Synchronous Serial Interface

6.1 Overview

The synchronous serial interface (SSI) is a DSP peripheral function to enable the communication between the CPU and the DSP.
It only supports synchronous serial communication and uses the DCLK (transmit/receive clock), DTXD (transmit data), and DRXD (receive data) signal lines.
Table 6.1 lists the specifications, and Figure 6.1 shows a block diagram of the synchronous serial interface.

Table 6.1 Specifications

Item	Specification
Master/slave	MCU/DSP
Communication mode	Synchronous (3-wire serial) DTXD: Transmit data DRXD: Receive data DCLK: Transmit/receive clock
Character length	8 bits
Interrupt generating timing	• Receive interrupt When data transferred from the receive shift register to the RBR register is completed • Transmit buffer empty interrupt When data transferred from the TBR register to the transmit shift register is completed

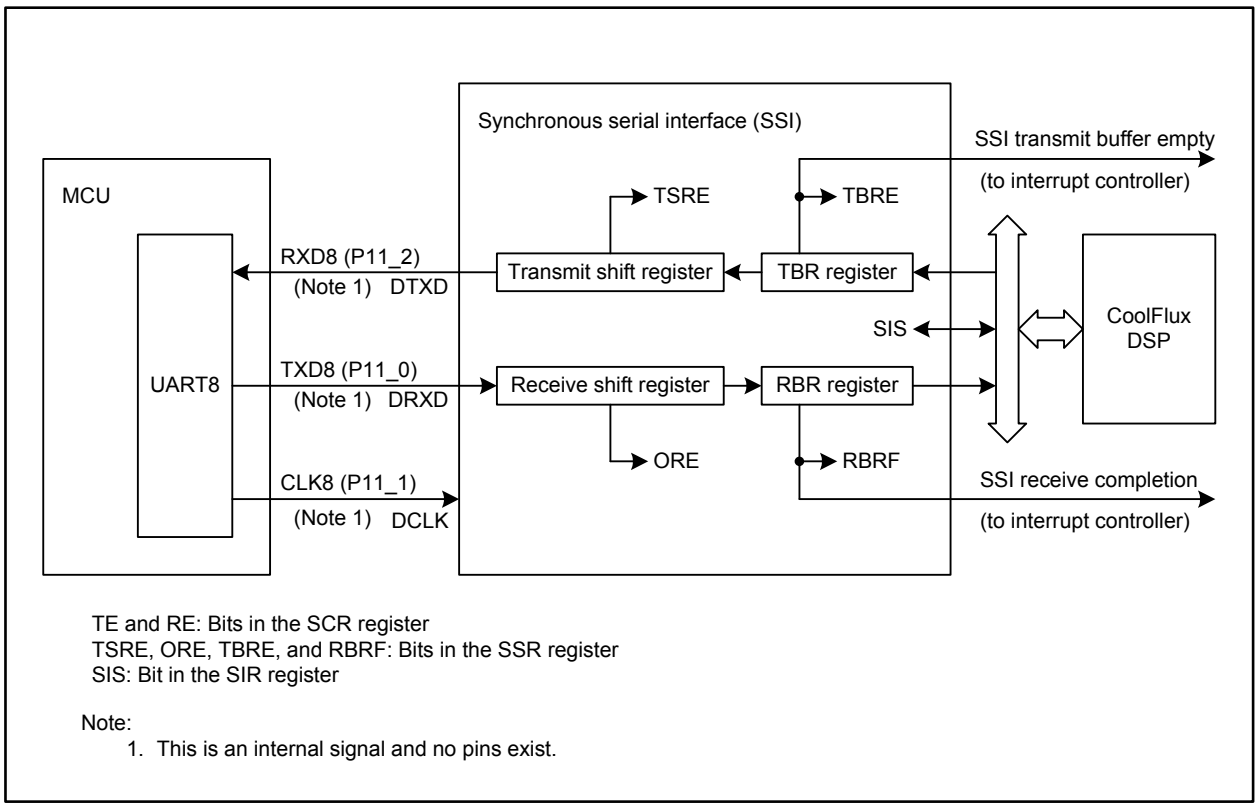


Figure 6.1 SSI Block Diagram

6.2 Registers

Table 6.2 lists functions of SSI associated registers, and Figure 6.2 to Figure 6.6 show the registers.

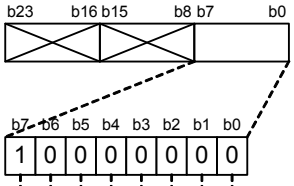
Table 6.2 Functions of Serial Communication Associated Registers

Address	Register	Symbol	Function
0020h	Serial Interface Setting Register	SIR	Enabling serial interface
0022h	Serial Control Register	SCR	Enabling/disabling communication
0023h	Transmit Buffer Register	TBR	Setting data to be transmitted to the MCU
0024h	Serial Status Register	SSR	Communication status
0025h	Receive Buffer Register	RBR	Data received from the MCU

The following DAP control registers also need to be set by the MCU:

- Set the TXDD bit in the DPC4 register to 1 (set the DTXD pin to output).
- Set the TXDS bit in the DPC10 register to 1 (output the DTXD signal).
- Set UART8 communication associated registers.

Serial Interface Setting Register

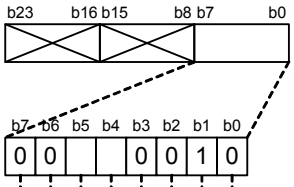


Symbol: SIR
Address: 0020h
Reset Value: 00 0000h

Bit Symbol	Bit Name	Function	RW
— (b6-b0)	Reserved	Should be written with 0	WO
SIS	Synchronous Serial Interface Enabled Bit	Should be written with 1	RW
— (b23-b8)	No register bits; should be written with 0 and read as 0		—

Figure 6.2 SIR Register

Serial Interface Control Register



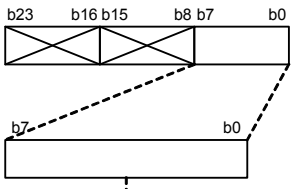
Symbol	Address	Reset Value
SCR	0022h	00 0000h

Bit Symbol	Bit Name	Function	RW
— (b0)	Reserved	Should be written with 0	RW
— (b1)	Reserved	Should be written with 1	RW
— (b3-b2)	Reserved	Should be written with 0	RW
RE	Receive Enable Bit (1)	0: Reception disabled 1: Reception enabled (2)	RW
TE	Transmit Enable Bit (1)	0: Transmission disabled 1: Transmission enabled (2)	RW
— (b7-b6)	Reserved	Should be written with 0	RW
— (b23-b8)	No register bits; should be written with 0 and read as 0		—

Notes:
1. Changes should be made after setting bits RE and TE to 0.
2. Enable the SSI with the SIS bit in the SIR register before setting this bit to 1.

Figure 6.3 SCR Register

Transmit Buffer Register

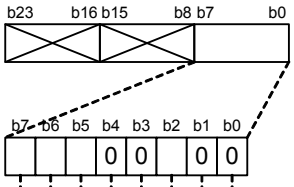


Symbol	Address	Reset Value
TBR	0023h	00 00XXh

Bit Symbol	Function	RW
— (b7-b0)	Data to be transmitted	WO
— (b23-b8)	No register bits; should be written with 0 and read as 0	—

Figure 6.4 TBR Register

Serial Status Register



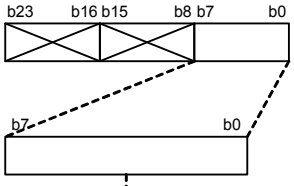
Symbol	Address	Reset Value
SSR	0024h	00 0084h

Bit Symbol	Bit Name	Function	RW
— (b1-b0)	Reserved	Should be written with 0	RW
TSRE	Transmit Shift Register Empty Flag	0: Data held in the transmit shift register (during transmission) 1: No data in the transmit shift register (transmission completed)	RO
— (b4-b3)	Reserved	Should be written with 0	RW
ORE	Overrun Error Flag (1)	0: No overrun error 1: Overrun error	RW
RBRF	Receive Complete Flag (1)	0: No data in the RBR register 1: Data held in the RBR register	RW
TBRE	Transmit Buffer Empty Flag (1)	0: Data held in the TBR register 1: No data in the TBR register	RW
— (b23-b8)	No register bits; should be written with 0 and read as 0		—

Note:
1. This bit can only be set to 0. It should not be set to 1.

Figure 6.5 SSR Register

Receive Buffer Register



Symbol	Address	Reset Value
RBR	0025h	00 00XXh

Bit Symbol	Function	RW
— (b7-b0)	Received data	RO
— (b23-b8)	No register bits; should be written with 0 and read as 0	—

Figure 6.6 RBR Register

6.3 Data Transmission and Reception

6.3.1 Data Transmission

When data is set in the TBR register while the transmit shift register is empty, the data is transferred to the transmit shift register and a transmission starts.

The double-buffered structure of the transmit buffer register and transmit shift register enables continuous data transmission. If the next transmit data has already been written to the TBR register when one character (8 bits) of data has been transmitted, the data is transferred to the transmit shift register and transmitted continuously.

Make sure that the TBRE bit in the SSR register is 1 (no data in the TBR register) to set data in the TBR register.

An SSI transmit buffer empty interrupt request is generated toward the interrupt controller in the DAP at the same time the TBRE bit becomes 1.

When a synchronous clock signal is input while the TE bit in the SCR register is 0 (transmission disabled), the following values are transmitted: FFh when the MSB of the previously transmitted data is 1, and 00h when the MSB is 0.

Perform the following transmission procedure:

- Before transmitting
  - (1) Set bits TE and RE in the SCR register to 0 (transmission/reception disabled).
  - (2) Set the SIS bit in the SIR register to 1.
  - (3) Set the TE bit in the SCR register to 1 (transmission enabled).
- Transmitting data
  - (4) Wait for the TBRE bit in the SSR register to become 1 (no data in the TBR register).
  - (5) Write transmit data to the TBR register.

Figure 6.7 shows an operation example of data transmission.

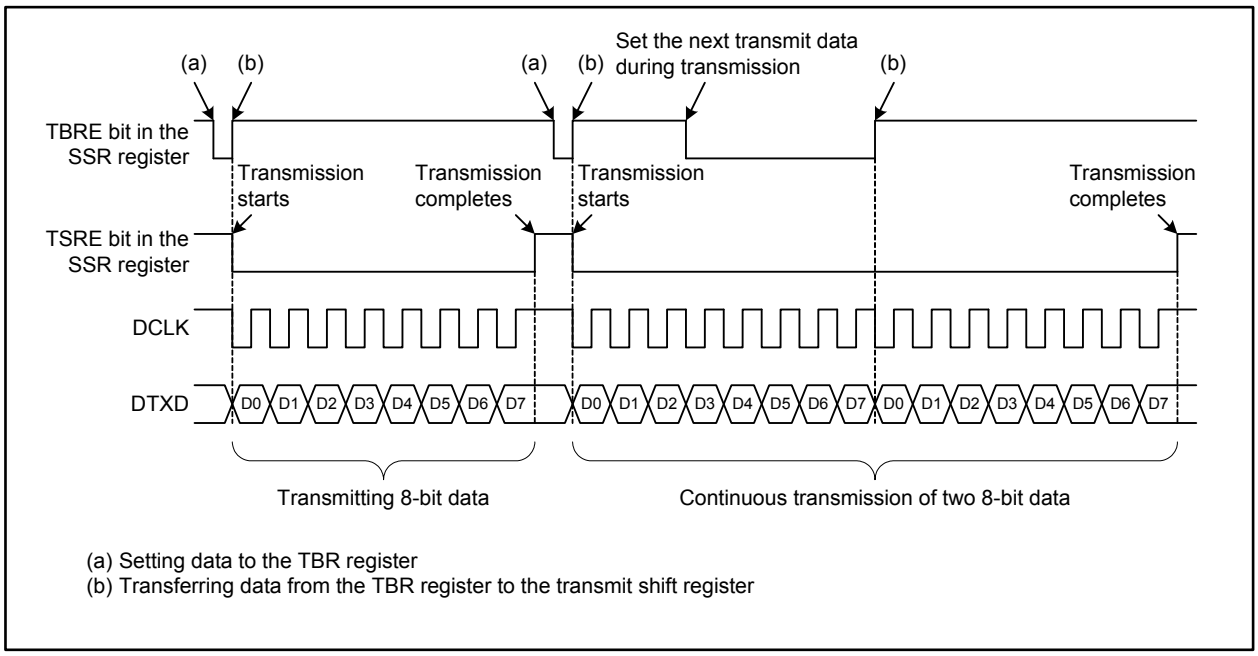

(a)
(b)
(a)
(b)
Set the next transmit data during transmission
(b)
TBRE bit in the SSR register
Transmission starts
Transmission completes
Transmission starts
Transmission completes
TSRE bit in the SSR register
DCLK
DTXD
D0 D1 D2 D3 D4 D5 D6 D7
D0 D1 D2 D3 D4 D5 D6 D7
D0 D1 D2 D3 D4 D5 D6 D7
Transmitting 8-bit data
Continuous transmission of two 8-bit data
(a) Setting data to the TBR register
(b) Transferring data from the TBR register to the transmit shift register

Figure 6.7 Operation Example of Data Transmission

6.3.2 Data Reception

When one character (8 bits) of data is received, the data is transferred from the receive shift register to the RBR register and the next data reception becomes enabled.
The double-buffered structure of the RBR register and receive shift register enables continuous data reception.
Make sure that the RBRF bit in the SSR register is 1 (data held in the RBR register) before reading the RBR register.
An SSI receive complete interrupt request is generated toward interrupt controller in the DAP at the same time as the RBRF bit becomes 1.

Perform the following reception procedure:

- Before receiving
  (1) Set bits TE and RE in the SCR register to 0 (transmission/reception disabled).
  (2) Set the SIS bit in the SIR register to 1.
  (3) Set the RE bit in the SCR register to 1 (reception enabled).
- Receiving data
  (4) Wait for the RBRF bit in the SSR register to become 1 (data held in the RBR register).
  (5) Read data from the TBR register.

Figure 6.8 shows an operation example of data reception.

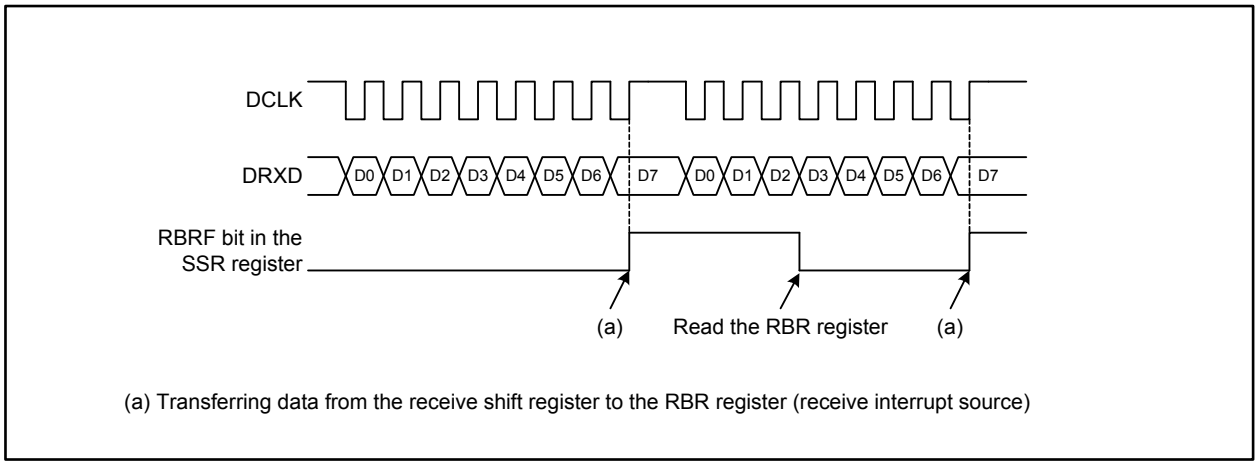


Figure 6.8 Operation Example of Data Reception

6.3.3 Overrun Error

An overrun error occurs when the next data is received before reading the RBR register. The ORE bit in the SSR register thereby becomes 1 (overrun error occurs).
Figure 6.9 shows the timing of an overrun error.

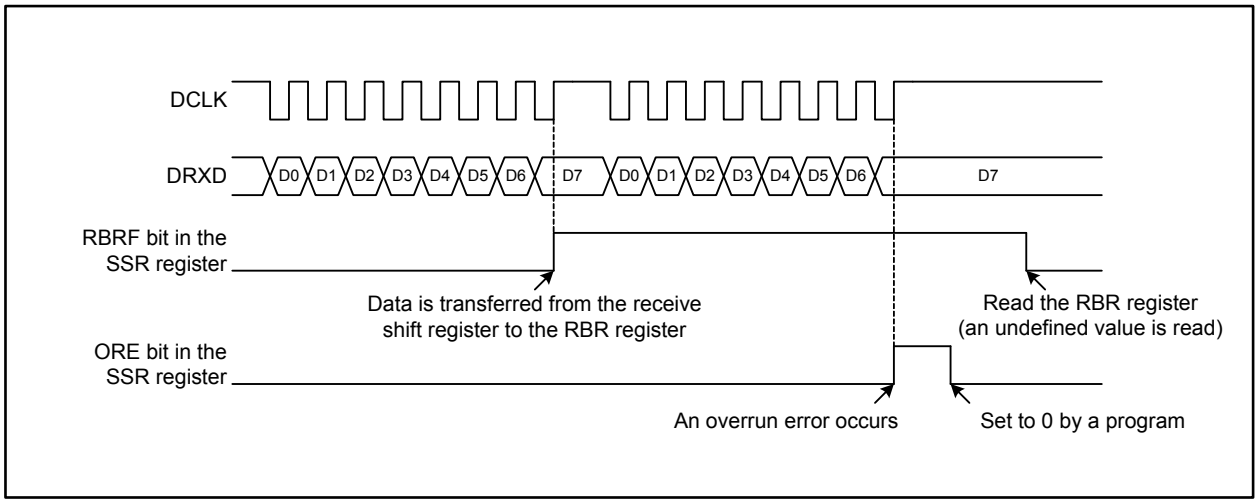


Figure 6.9 Overrun Error Timing

When an overrun error occurs, set the ORE bit to 0 by a program and then read the RBR register to enable the next data reception. The read value is undefined.

7. Interrupt Controller

7.1 Overview

The DAP has an internal interrupt controller aside from the one in the MCU. Interrupt requests to the DSP are transmitted from this internal interrupt controller.

There are two types of interrupts: the SAI interrupt, which is generated by the audio interface at each frame, and the SSI interrupt, which is generated by the SSI at transmission/reception. Either the reception complete interrupt or transmission buffer empty interrupt can be selected for the SSI interrupt.

Note that the SAI interrupt request has a higher priority and is accepted first when the SAI interrupt request and SSI interrupt request are generated simultaneously.

Table 7.1 lists interrupts, and Figure 7.1 shows a block diagram of the interrupt controller.

Table 7.1 Interrupts

Interrupt		Interrupt Request Generation Timing
SAI interrupt		At every frame (1.5 cycles of the BTCLK clock + 1 to 2 cycles of the DSP clock after the rising edge (1) of the LRCLK signal)
SSI interrupt (2)	SSI reception complete interrupt	When the RBRF bit in the SSR register changes from 0 to 1
	SSI transmit buffer empty interrupt	When the TBRE bit in the SSR register changes from 0 to 1

Notes:
1. It is the falling edge when using the I²S format.
2. Either the SSI reception complete interrupt or SSI transmit buffer empty interrupt can be selected for the SSI interrupt.

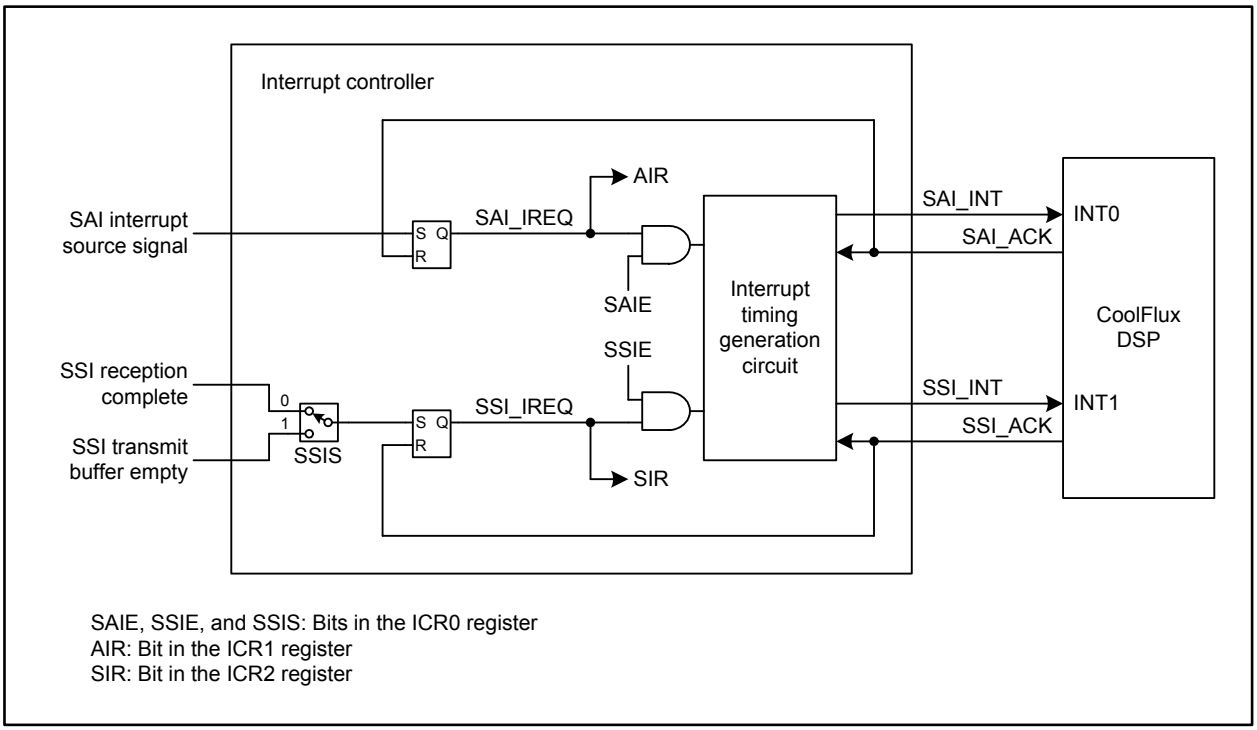
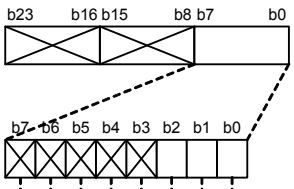


Figure 7.1 Interrupt Controller Block Diagram

Signals SAI_INT and SSI_INT are connected to DSP interrupt port 0 (INT0) and interrupt port 1 (INT1), respectively.
When an interrupt is accepted, the instruction jumps to the address set in the interrupt vector table. Refer to “CoolFlux DSP Assembly Programmer’s Manual” or “CoolFlux DSP C Programmer’s Manual” for details on the interrupt vectors.
Figure 7.2 to Figure 7.4 show the interrupt control registers.

7.2 Registers

Interrupt Control Register 0

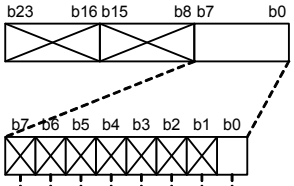


Symbol	Address	Reset Value
ICR0	0000h	00 0000h

Bit Symbol	Bit Name	Function	RW
SAIE	SAI Interrupt Enable Bit	0: Interrupt disabled 1: Interrupt enabled	RW
SSIE	SSI Interrupt Enable Bit	0: Interrupt disabled 1: Interrupt enabled	RW
SSIS	SSI Interrupt Select Bit	0: SSI reception complete interrupt 1: SSI transmit buffer empty interrupt	RW
— (b23-b3)	No register bits; should be written with 0 and read as undefined value		—

Figure 7.2 ICR0 Register

Interrupt Control Register 1



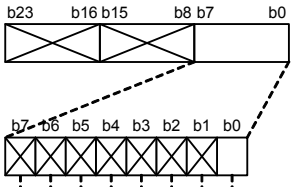
Symbol	Address	Reset Value
ICR1	0001h	00 0000h

Bit Symbol	Bit Name	Function	RW
AIR	SAI Interrupt Request Flag (1)	0: No interrupt requested 1: Interrupt requested	RW
— (b23-b3)	No register bits; should be written with 0 and read as undefined value		—

Note:
1. This bit automatically becomes 0 once the DSP accepts an interrupt when the SAIE bit in the ICR0 register is 1 (interrupt enabled). Do not set this bit to 0 by a program. Although this bit becomes 1 when an interrupt is generated even when the SAIE bit is 0 (interrupt disabled), it does not automatically revert back to 0 since the DSP does not accept the interrupt. In this case, set it to 0 by a program.

Figure 7.3 ICR1 Register

Interrupt Control Register 2



Symbol	Address	Reset Value
ICR2	0002h	00 0000h

Bit Symbol	Bit Name	Function	RW
SIR	SSI Interrupt Request Flag (1)	0: No interrupt requested 1: Interrupt requested	RW
— (b23-b3)	No register bits; should be written with 0 and read as undefined value		—

Note:

1. This bit automatically becomes 0 once the DSP accepts an interrupt when the SSIE bit in the ICR0 register is 1 (interrupt enabled). Do not set this bit to 0 by a program. Although this bit becomes 1 when an interrupt is generated even when the SSIE bit is 0 (interrupt disabled), it does not automatically revert back to 0 since the DSP does not accept the interrupt. In this case, set it to 0 by a program.

Figure 7.4 ICR2 Register

7.3 Operation Sequence

7.3.1 Interrupt Setting Procedure

• Steps to set the SAI interrupt
(1) Set the SAIE bit in the ICR0 register to 0 (interrupt disabled).
(2) Set the AIR bit in the ICR1 register to 0 (no interrupt requested).
(3) Wait for the AIR bit to become 1 (interrupt requested).
(4) Set the AIR bit to 0.
(5) Set the SAIE bit to 1 (interrupt enabled).

• Steps to set the SSI reception complete interrupt
(1) Set the SSIE bit in the ICR0 register to 0 (interrupt disabled).
(2) Set the SIR bit in the ICR2 register to 0 (no interrupt requested).
(3) Set the SSIS bit in the ICR0 register to 0 (SSI reception complete interrupt).
(4) Set the SSIE bit to 1 (interrupt enabled).

• Steps to set the SSI transmit buffer empty interrupt
(1) Set the SSIE bit in the ICR0 register to 0 (interrupt disabled).
(2) Set the SIR bit in the ICR2 register to 0 (no interrupt requested).
(3) Set the SSIS bit in the ICR0 register to 1 (SSI transmit buffer empty interrupt).
(4) Set the SSIE bit to 1 (interrupt enabled).

7.3.2 Single Interrupt

Figure 7.5 shows an operation example of when a single interrupt is generated.

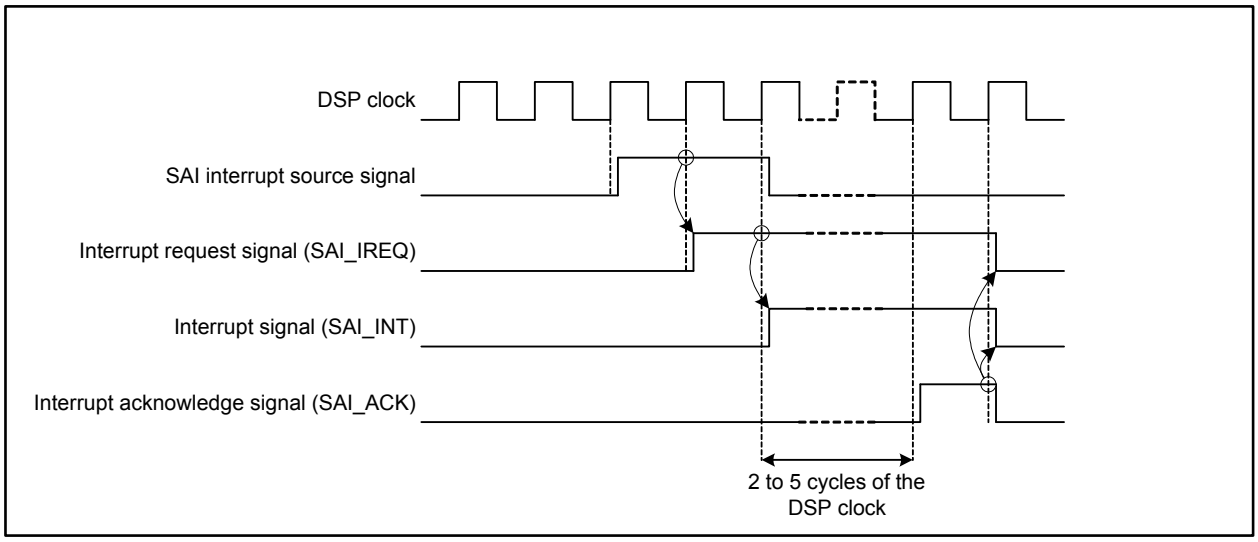

DSP clock
SAI interrupt source signal
Interrupt request signal (SAI_IREQ)
Interrupt signal (SAI_INT)
Interrupt acknowledge signal (SAI_ACK)
2 to 5 cycles of the DSP clock

Figure 7.5 Operation Example of a Single Interrupt

7.3.3 Multiple Interrupts

Figure 7.6 shows an operation example when multiple interrupts are generated.

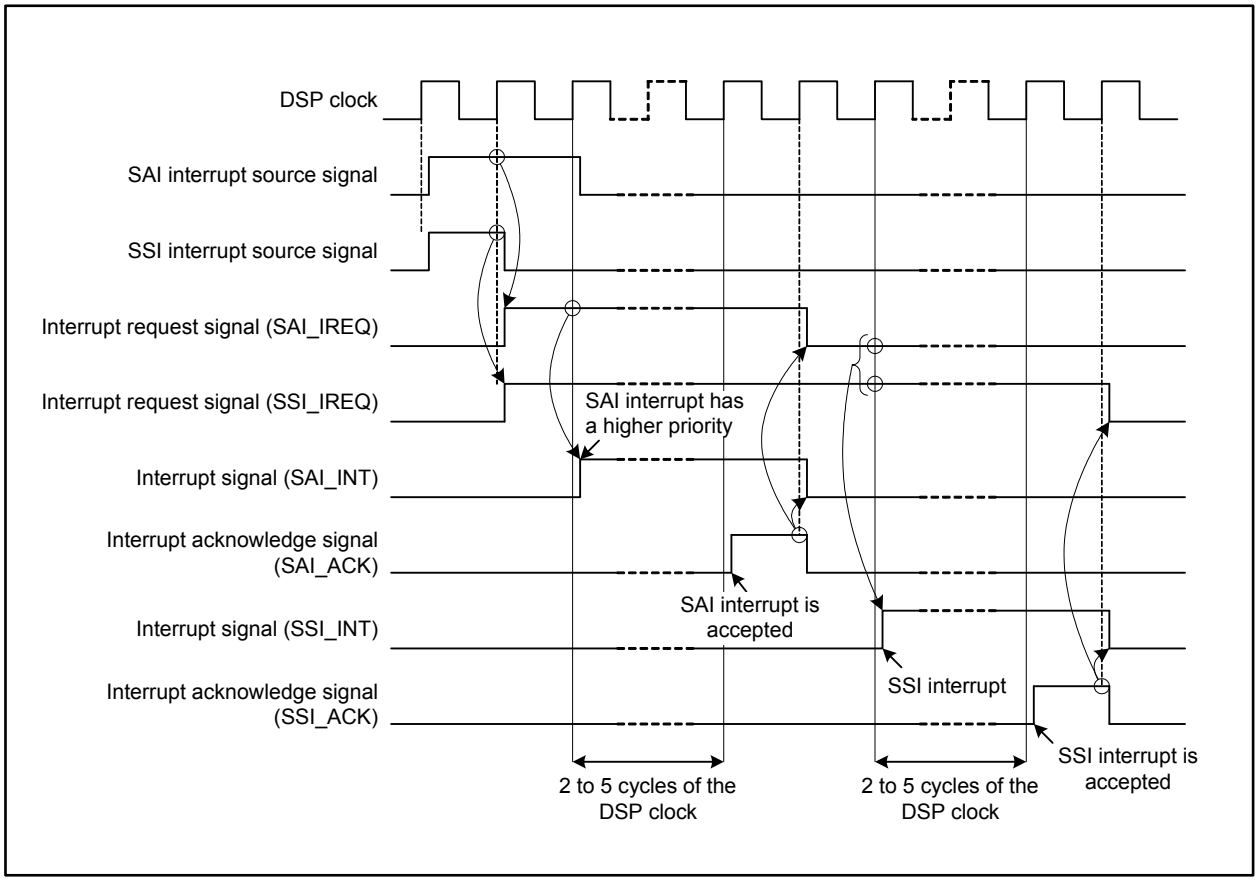

DSP clock
SAI interrupt source signal
SSI interrupt source signal
Interrupt request signal (SAI_IREQ)
Interrupt request signal (SSI_IREQ)
Interrupt signal (SAI_INT)
Interrupt acknowledge signal (SAI_ACK)
Interrupt signal (SSI_INT)
Interrupt acknowledge signal (SSI_ACK)
SAI interrupt has a higher priority
SAI interrupt is accepted
SSI interrupt
SSI interrupt is accepted
2 to 5 cycles of the DSP clock
2 to 5 cycles of the DSP clock

Figure 7.6 Operation Example of Multiple Interrupts

8. General Purpose I/O Ports

8.1 Overview

General purpose I/O ports are connected to MCU port P10 in a wired-OR configuration. They enable communication between the DSP and MCU, and communication with external devices using ports DP_0 to DP_7.

Use the CPU programs to set the general purpose I/O ports. The ports can be set to either input or output in 1-bit units. Refer to 8.3 “Input/Output Setting” for details.

Figure 8.1 shows a block diagram associated with the general purpose ports.

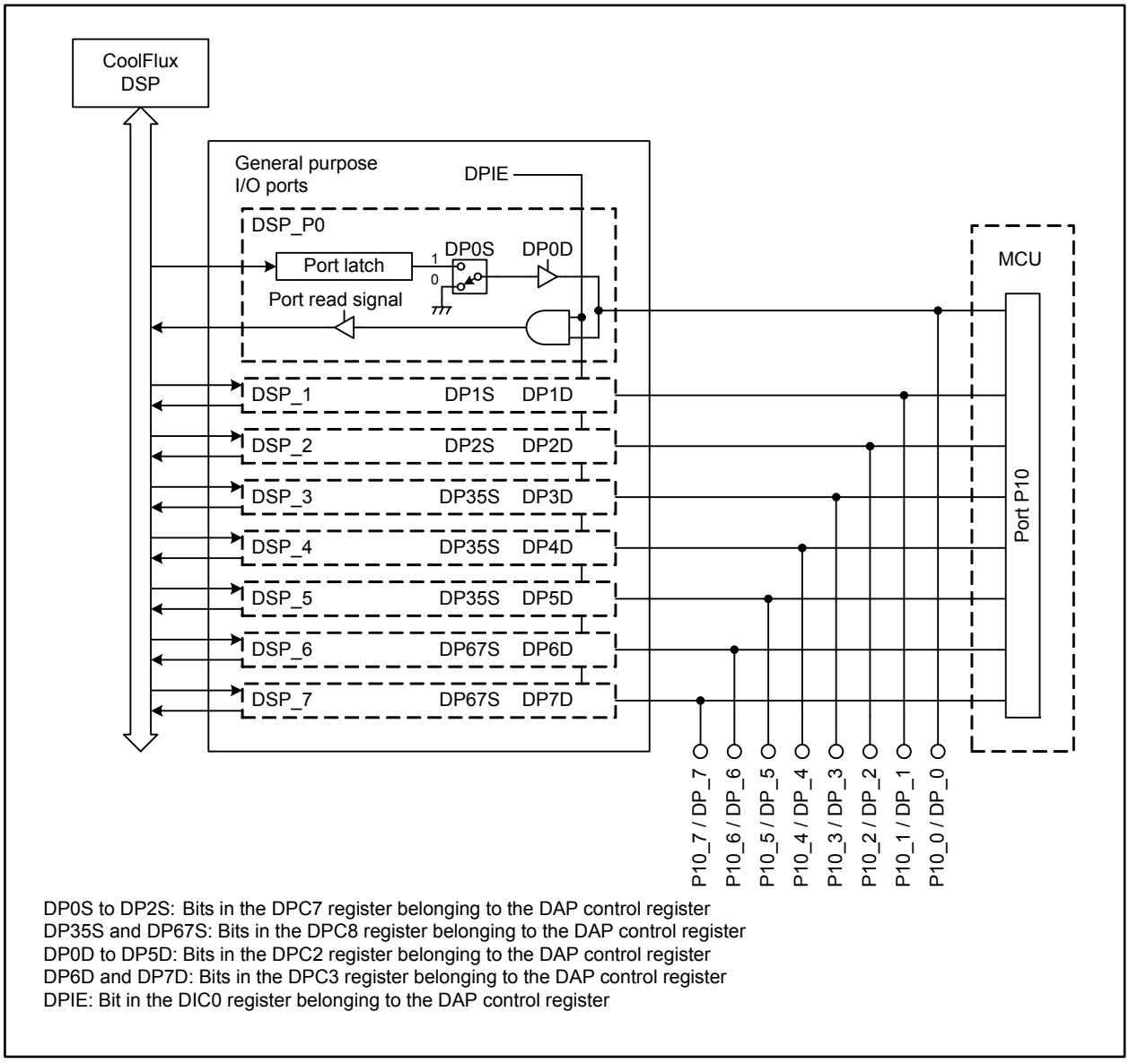
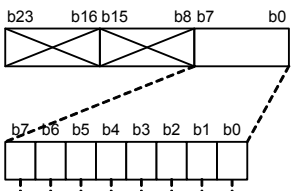

CoolFlux DSP
General purpose I/O ports
DPIE
DSP_P0
Port latch
DP0S
DP0D
Port read signal
DSP_1 DP1S DP1D
DSP_2 DP2S DP2D
DSP_3 DP35S DP3D
DSP_4 DP35S DP4D
DSP_5 DP35S DP5D
DSP_6 DP67S DP6D
DSP_7 DP67S DP7D
MCU
Port P10
P10_7 / DP_7
P10_6 / DP_6
P10_5 / DP_5
P10_4 / DP_4
P10_3 / DP_3
P10_2 / DP_2
P10_1 / DP_1
P10_0 / DP_0
DP0S to DP2S: Bits in the DPC7 register belonging to the DAP control register
DP35S and DP67S: Bits in the DPC8 register belonging to the DAP control register
DP0D to DP5D: Bits in the DPC2 register belonging to the DAP control register
DP6D and DP7D: Bits in the DPC3 register belonging to the DAP control register
DPIE: Bit in the DIC0 register belonging to the DAP control register

Figure 8.1 General Purpose I/O Port Block Diagram

8.2 Register

DSP I/O Register



Symbol	Address	Reset Value
DIOR	0007h	00 0000h

Bit Symbol	Bit Name	Function	RW
DP_0	General Purpose I/O Port 0	When set as input A value is written to the corresponding bit. It is not output to DAP general purpose I/O ports due to input mode selected. The read value is the state of the corresponding DAP general purpose I/O ports as follows: 0: Low 1: High When set as output The written value is reflected to the level of the corresponding general purpose I/O port. 0: Output low 1: Output high The read value is the value set in the corresponding bit.	RW
DP_1	General Purpose I/O Port 1		RW
DP_2	General Purpose I/O Port 2		RW
DP_3	General Purpose I/O Port 3		RW
DP_4	General Purpose I/O Port 4		RW
DP_5	General Purpose I/O Port 5		RW
DP_6	General Purpose I/O Port 6		RW
DP_7	General Purpose I/O Port 7		RW
— (b23-b8)	No register bits; should be written with 0 and read as 0		—

Figure 8.2 DIOR Register

8.3 Input/Output Setting

Set pin I/O using the CPU programs.

8.3.1 Input Setting

Perform the following to set the general purpose I/O ports to input and MCU port P10 to input/output.

(1) Setting input or output direction of the general purpose I/O ports
Set the corresponding bits in registers DPC2 and DPC3 belonging to the DAP control register to 0 (input).

(2) Setting input or output direction of MCU port P10
- Inputting a signal from pins DP_0 to DP_7 to the general purpose I/O ports
  Set the corresponding bit in the PD10 register (SFR) to 0 (input).
- Inputting output data of the MCU port P10 to the general purpose I/O ports
  Set the corresponding bit in the PD10 register (SFR) to 1 (output).

(3) Enabling a signal input of pins DP_0 to DP_7 to the general purpose I/O ports
Set the DPE bit in the DIC0 register belonging to the DAP control register to 1 (DP_0 to DP_7 signal input enabled).

Make sure that ports P10_0 to P10_7 and the corresponding ports DP_0 to DP_7 are not set to output at the same time. Figure 8.3 shows input signal flows according to the I/O setting.

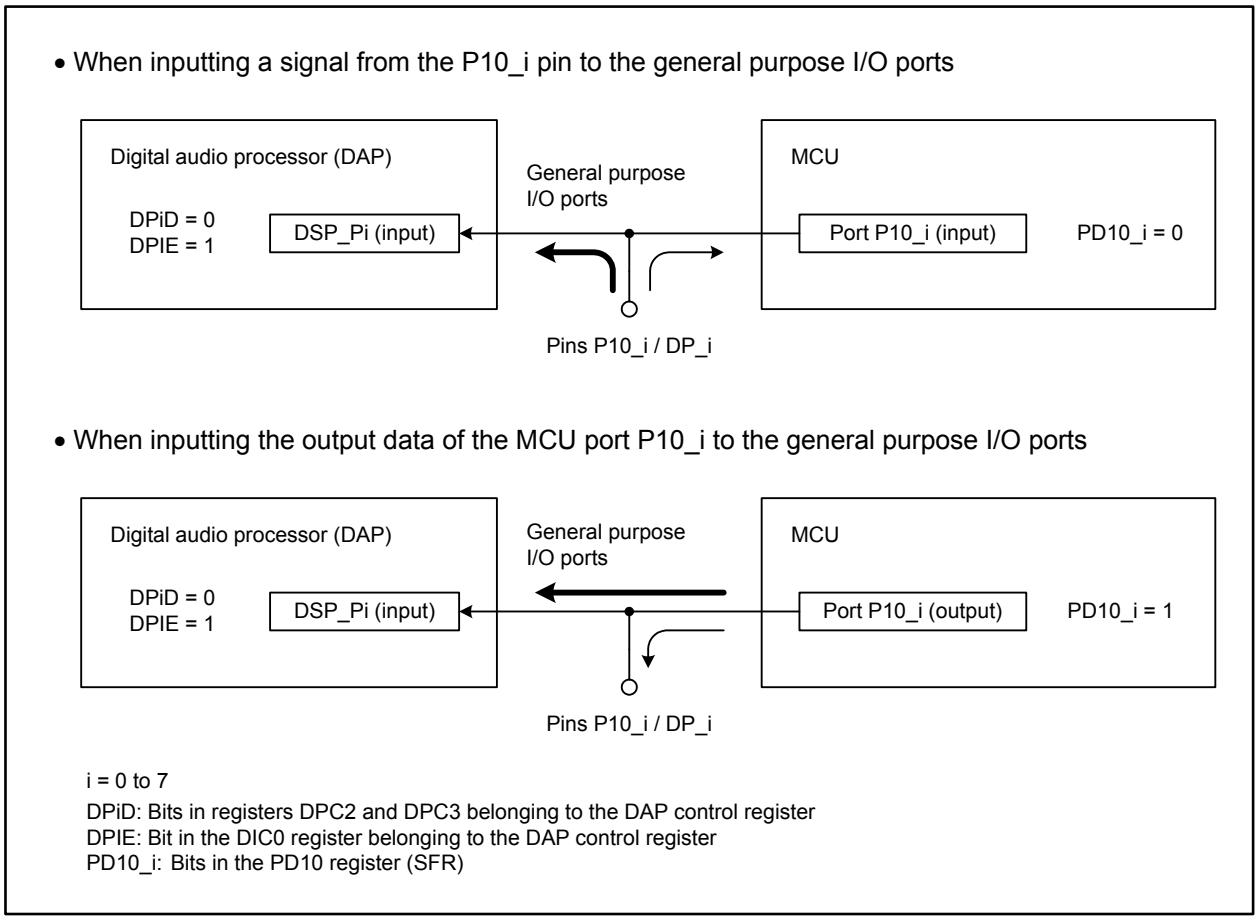

• When inputting a signal from the P10_i pin to the general purpose I/O ports
Digital audio processor (DAP)
DPiD = 0
DPIE = 1
DSP_Pi (input)
General purpose I/O ports
MCU
Port P10_i (input)
PD10_i = 0
Pins P10_i / DP_i
• When inputting the output data of the MCU port P10_i to the general purpose I/O ports
Digital audio processor (DAP)
DPiD = 0
DPIE = 1
DSP_Pi (input)
General purpose I/O ports
MCU
Port P10_i (output)
PD10_i = 1
Pins P10_i / DP_i
i = 0 to 7
DPiD: Bits in registers DPC2 and DPC3 belonging to the DAP control register
DPIE: Bit in the DIC0 register belonging to the DAP control register
PD10_i: Bits in the PD10 register (SFR)

Figure 8.3 Input Signal Flows

8.3.2 Output Setting

Perform the following to set the general purpose I/O ports to output and MCU port P10 to input:

(1) Set the corresponding bit in the PD10 register (SFR) to 0 (input).
(2) Set the corresponding bits in registers DPC2 and DPC3 belonging to the DAP control register to 1 (output).
(3) Set the corresponding bits in registers DPC7 and DPC8 belonging to the DAP control register to 1 (output the general purpose I/O port signal).
(4) Set the DPIE bit in the DIC0 register belonging to the DAP control register to 1 (DP_0 to DP_7 signal input enabled) (only when reading the output data).

Make sure the corresponding bits of the both ports are not set to output at the same time. Figure 8.4 shows an output signal flow according to the I/O setting.

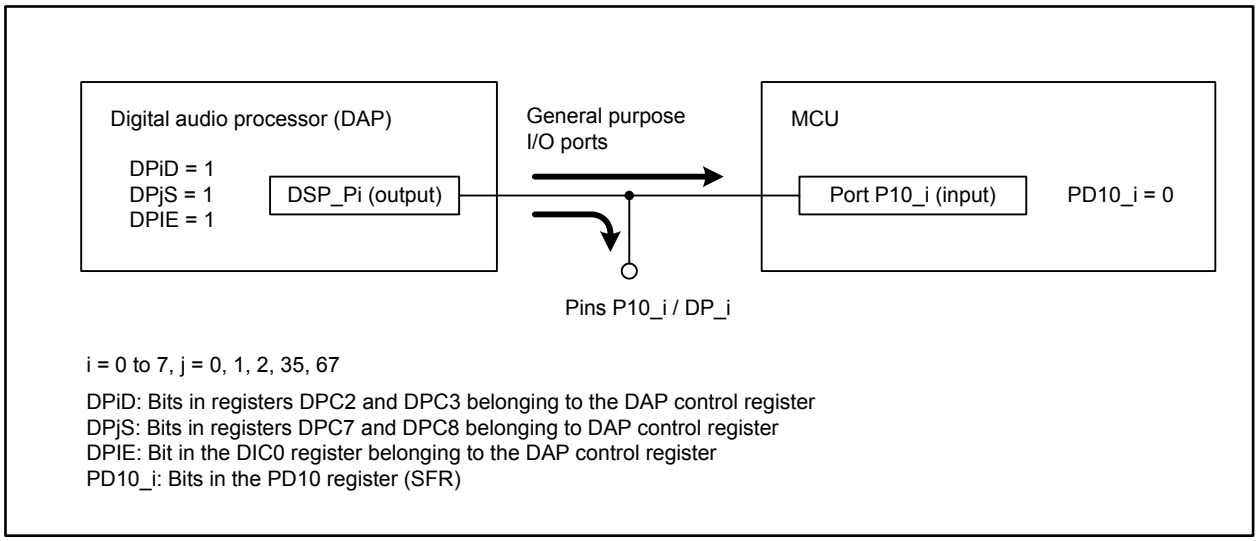

Digital audio processor (DAP)
DPiD = 1
DPjS = 1
DPIE = 1
DSP_Pi (output)
General purpose
I/O ports
MCU
Port P10_i (input)
PD10_i = 0
Pins P10_i / DP_i
i = 0 to 7, j = 0, 1, 2, 35, 67
DPiD: Bits in registers DPC2 and DPC3 belonging to the DAP control register
DPjS: Bits in registers DPC7 and DPC8 belonging to DAP control register
DPIE: Bit in the DIC0 register belonging to the DAP control register
PD10_i: Bits in the PD10 register (SFR)

Figure 8.4 Output Signal Flow

9. JTAG Interface

An on-chip emulator is embedded in the CoolFlux DSP, and debugging is controlled externally via a JTAG interface.
Refer to “CoolFlux DSP Multi-core Debugger User Guide” for details.

10. Bootloader

The DSP operates according to the DSP programs stored in the DSP program RAM. When the power is turned on, the DSP programs need to be transferred from the MCU ROM to the DSP program RAM since no programs are stored in the program RAM. The bootloader, which is stored in the DSP boot ROM as a default, is a program for storing DSP programs and data transferred from the MCU to the DSP RAMs.
This chapter explains the procedure from the DSP program transfer to the start-up.

10.1 Overview

The DSP supports program transfer mode and program execution mode. They can be selected with the DSPC register belonging to the DAP control register using the MCU programs. The DSP starts in program transfer mode by setting the DMOD bit in the DSPC register to 0 and then the DRR bit from 0 (reset) to 1 (reset released), and the bootloader runs. After the program transfer, the DSP starts in program execution mode by setting the DMOD bit to 1 and then the DRR bit from 0 to 1, and the DSP programs stored in the program RAM are executed.

10.2 Procedure on Starting DSP

The MCU needs to transfer the DSP programs to DSP memories in order to start the DSP.
Figure 10.1 shows the procedure for the MCU programs to be executed, from transferring the DSP programs to starting the DSP.

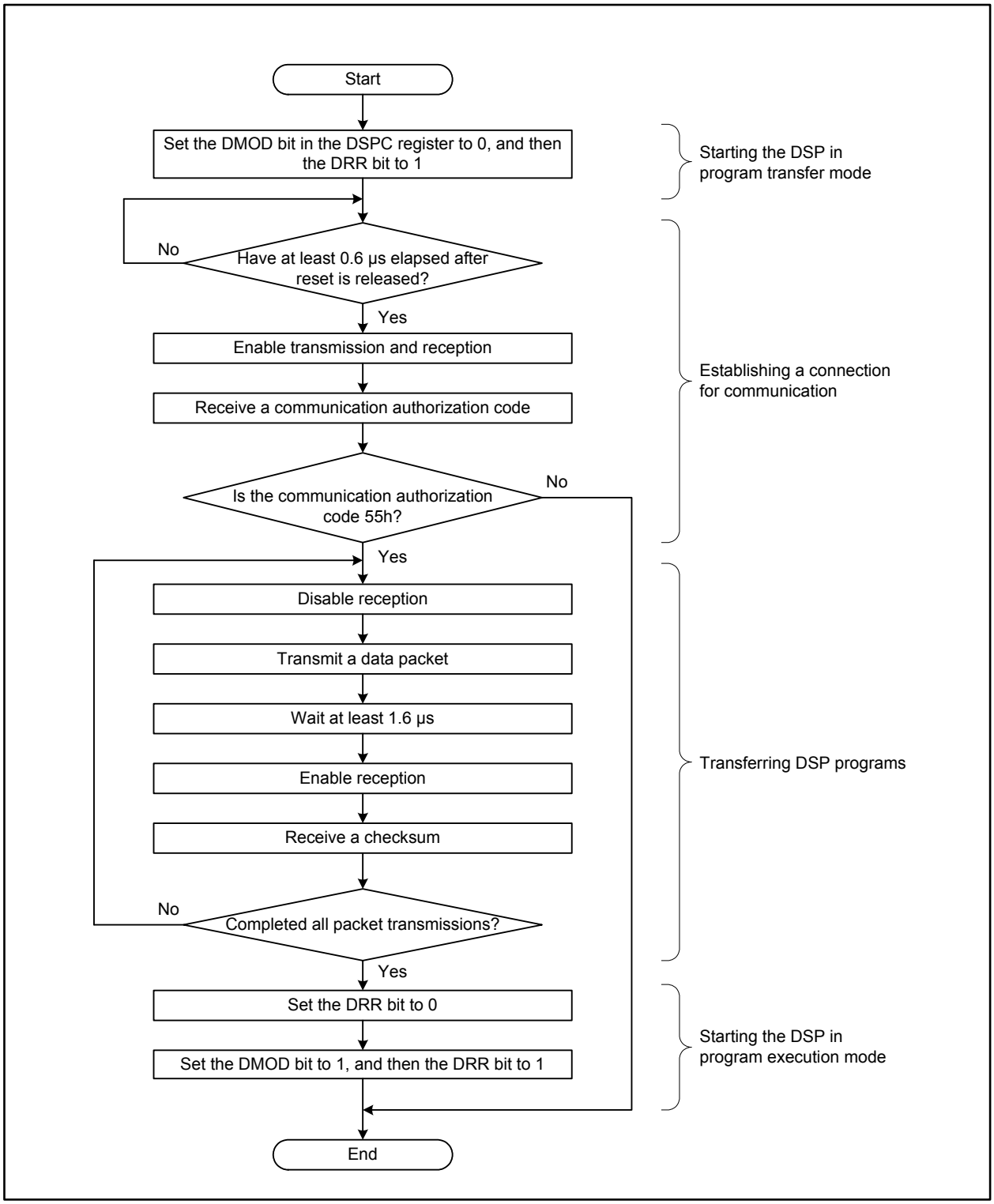

Start
Set the DMOD bit in the DSPC register to 0, and then the DRR bit to 1
Starting the DSP in program transfer mode
No
Have at least 0.6 µs elapsed after reset is released?
Yes
Enable transmission and reception
Receive a communication authorization code
Establishing a connection for communication
Is the communication authorization code 55h?
No
Yes
Disable reception
Transmit a data packet
Wait at least 1.6 µs
Enable reception
Receive a checksum
Transferring DSP programs
No
Completed all packet transmissions?
Yes
Set the DRR bit to 0
Set the DMOD bit to 1, and then the DRR bit to 1
Starting the DSP in program execution mode
End

Figure 10.1 DSP Start-up Procedure (MCU Programs)

Establishing a connection for the communication is required in order to enable the communication between the MCU programs and the bootloader. After setting the DRR bit in the DSPC register from 0 (reset) to 1 (reset released), wait at least 0.6 µs to receive a communication authorization code, and make sure that the code is 55h.
When the communication authorization code is not 55h, reset the DSP again since the connection for the communication is not yet established.
Refer to 10.5 “Communication Authorization Code and Checksum Reception” for details on receiving the communication authorization code.
Refer to 10.3 “DSP Program Transfer” for details on transferring the DSP programs.

10.3 DSP Program Transfer

The MCU transfers the DSP programs in packet units. After transmitting a data packet, the MCU waits at least 1.6 µs to receive a checksum. This is repeated when transmitting multiple data packets.

10.3.1 Data Packet

A data packet consists of a packet header, data length, and transfer data. Data is transferred in word units whose data length is either 32 bits or 24 bits. The number of words which can be transferred per packet is limited. Thus, when that limit is exceeded, the data needs to be divided into multiple packets to be transferred. Refer to 10.3.1.2 “Data Length” for the number of transferable words per packet.
Figure 10.2 shows the data packet structure.

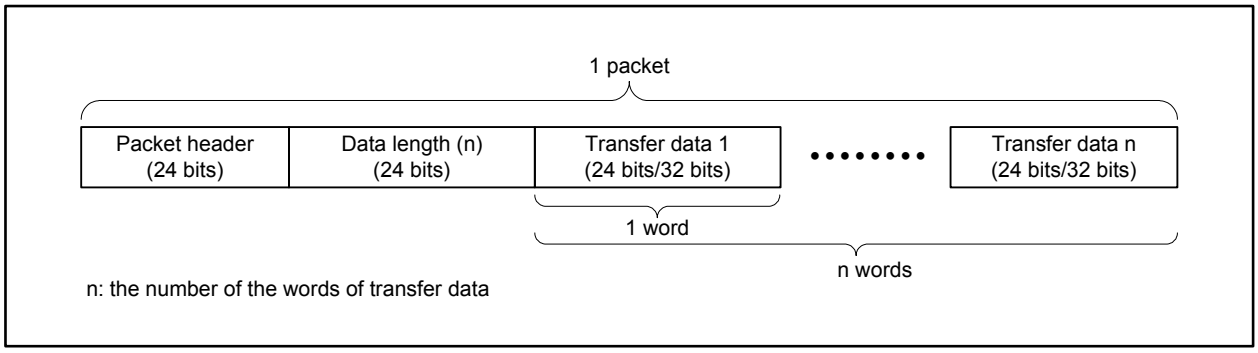

1 packet
Packet header (24 bits)
Data length (n) (24 bits)
Transfer data 1 (24 bits/32 bits)
Transfer data n (24 bits/32 bits)
1 word
n words
n: the number of the words of transfer data

Figure 10.2 Data Packet Structure

10.3.1.1 Packet Header

A packet header is 24-bit data to specify the memory type and start address of the destination. Table 10.1 lists its details.

Table 10.1 Packet Header Details

Bit	Bit Name	Bit Setting
b23	Packet type	Set this bit to 0
b22	Read/write bit	Set this bit to 0 (write)
b21 to b20	Memory	00b: Do not use this combination 01b: Program RAM 10b: Data RAM 11b: Coefficient RAM
b19 to b16	Reserved	Set these bits to 0
b15 to b0	Start address (1)	Specify the start address of a transfer in 16 bits

Note:
1. Address 0000h is reserved and should not be specified as the start address when writing to the data RAM.

10.3.1.2 Data Length

Data length specifies the number of words of transferring data in 24 bits. Data needs to be divided into multiple packets when the total number of words exceeds the range listed in Table 10.2.

Table 10.2 Setting Range of Data Words

Memory	Setting Range
Program RAM	1 to 2047
Data RAM and Coefficient RAM	1 to 4095

10.3.1.3 Transfer Data

Transfer data is the data itself to be transferred. The number of bits per word is 32 bits for the program RAM and 24 bits for the data RAM and coefficient RAM. The MCU repeatedly transmits the data for the number of times specified in the data length.

10.3.2 Checksum

The DSP calculates a checksum after receiving a packet data and returns the value to the MCU upon the request.

The checksum value is a lower byte of the total value, which is calculated by adding one packet of data (packet headers, data length, and all transfer data) together in 1-byte units.

The MCU should wait until a checksum calculation is completed (1.6 µs) before receiving a checksum.

Refer to 10.5 “Communication Authorization Code and Checksum Reception” for details on receiving a checksum.

10.4 Transmit Data Formats

Data in a packet is transmitted from upper to lower bytes, from the LSB. There is no need to insert waits between bytes.
Figure 10.3 shows the transmit data format.

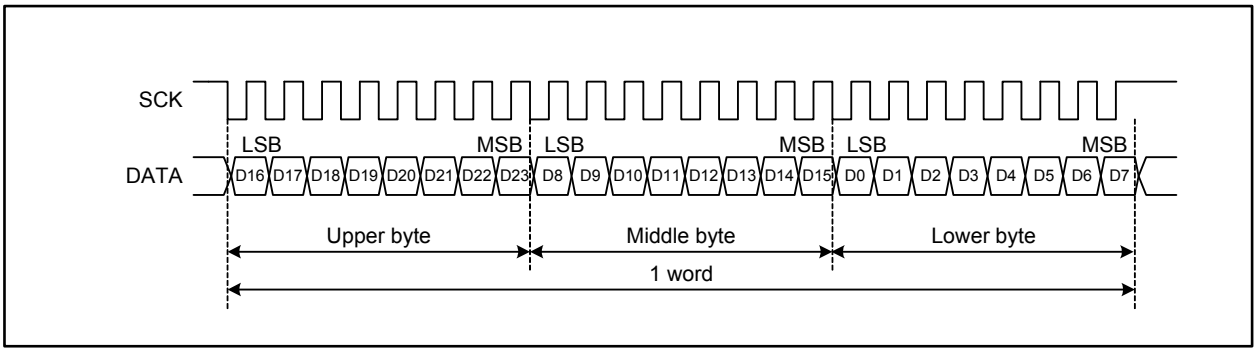


Figure 10.3 Transmit Data Format Using 24-bit Data

10.5 Communication Authorization Code and Checksum Reception

The MCU must generate clocks for one byte since synchronous serial communication, whose master is the MCU, is used to receive a communication authorization code and checksum.
The clock is generated when writing a dummy data to the transmit buffer. The dummy data value must be 00h.

Revision History	R32C/102 Group User's Manual: DSP

Rev.	Date	Description	
		Page	Summary
1.00	May 12, 2011	—	Initial release

R32C/102 Group User's Manual: DSP

Publication Date: Rev. 1.00 May 12, 2011

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2011 Renesas Electronics Corporation. All rights reserved.

Colophon 1.0

R32C/102 Group

Renesas Electronics Corporation

R01UH0208EJ0100